

Exhibit 6

UNITED STATES DISTRICT COURT
NORTHERN DISTRICT OF CALIFORNIA
SAN FRANCISCO DIVISION

ORACLE AMERICA, INC.,

Plaintiff,

v.

GOOGLE INC.,

Defendant.

Case No. CV 10-03561 WHA

**EXPERT REPORT OF ROBERT
ZEIDMAN**

```
\API23_Marshmallow\libcore\luni\src\main\java\javax\  
\API23_Marshmallow\libcore\libart\src\main\java\java\
```

29. These versions of Android analyzed and relied upon include those related to Google's Android Auto, Android TV, and Android Wear initiatives. Android apps designed for either Android Auto or Android TV must be compatible with, at a minimum, Android Lollipop (API Level 21). <http://developer.android.com/training/auto/start/index.html>; <http://developer.android.com/training/tv/start/start.html>. Android apps designed for Android Wear must be compatible with, at a minimum, Android Kit Kat (API Level 20). <http://developer.android.com/training/wearables/apps/creating.html>.

30. Absent from this list of Android source code I relied upon is the source code for Android Honeycomb (API Level 11 through 13). I was unable to download the source code from Google's online source code repository and I understand it is not available for download. <https://source.android.com/source/build-numbers.html#honeycomb-gpl-modules>. I was also unable to retrieve the source code from a hard drive Google produced in this case that purports to be a mirror of Google's Android Git repository because it appears files were missing that were required to use the Git repository. Instead of identifying copying of Java SE 5 declaring code in Android Honeycomb by analyzing source code, I achieved the same result using Google's own public statements. *See* Paragraph 108 through 111, below.

C. Software Tools

1. *Software Tools*

31. I relied upon numerous software tools as part of my analysis, including:

a. Understand

32. Understand is a commercial tool from Scientific Toolworks that analyzes and provides pertinent information regarding a collection of code. Specifically, Understand identifies all entities, which include fields, methods, constructors, parameters, classes, interfaces,

and annotations, and creates a Data Dictionary Report that can be used for further analysis. I used Understand to identify and output potential declaring code within both the Java and Android code bases that I reviewed. I also manually searched for package and import statements.

b. Cygwin

33. Cygwin is open-source software that creates a Linux command line environment in Windows. I used Cygwin to run Python scripts.

c. Python

34. Python is a widely used open source scripting language that has many built-in functions for parsing, searching, and managing strings. I used Python to compare source code from Java and Android.

III. BACKGROUND

35. This section describes the technological issues relevant as background to understand my analysis and opinion.

A. Software Source Code

36. Computer programs can be written using complex instructions that look like English. For example, the instruction `a = b*c+2` tells the computer to take the number stored in memory and represented by variable `b`, multiply that by the number stored in memory and represented by the variable `c`, add 2 and store the result in memory represented by the variable `a`. Similarly, the statement `printf("Hello world!")` tells the computer to print the words "Hello world!" to the computer screen. These high-level, English-like instructions are called "source code." Computer programs are made up of many lines of source code and the process of writing these lines of code is called programming. Eventually these lines of source code are turned into instructions that a computer understands, consisting of sequences of electronic ones and zeroes. The process of turning human-readable source code into a file containing computer instructions